

Traduction et adaptation des infos du site : <https://tronixlabs.com.au/news/tutorial-l298n-dual-motor-controller-module-2a-and-arduino/>

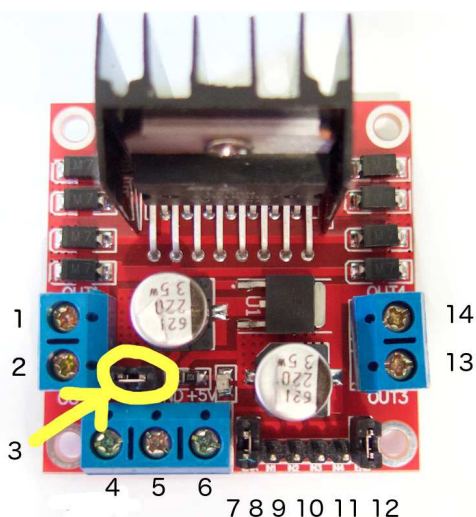
Notamment ce double pont en H permet le branchement de deux moteurs continus ou un moteur pas à pas 2 phases.

Ces moteurs peuvent être contrôlés aussi bien en vitesse et en direction à l'aide de sortie PWM.

Spécification:

- Alimentation de puissance: de +6V à +35V
- Tension de commande logique Vss: de +5 à +7V (alimentation interne de +5V)
- Courant Max : 2A
- Courant de commande logique: de 0 à 36mA
- Tensions pour contrôle de direction: Niveau bas -0.3V à 1.5V, niveau haut: 2.3V à Vss
- Tensions pour contrôle "Enable": Niveau bas -0.3V à 1.5V, niveau haut : 2.3V à Vss
- Puissance Max: 25W (Température 75 °C dissipateur thermique intégré)
- Température de fonctionnement: de -25°C à +130°C

Repère des bornes du module :



1	Sortie 1 Moteur DC "+" ou moteur pas à pas A+	
2	Sortie 1 Moteur DC "-" moteur pas à pas A-	
3	12V cavalier – A supprimer si vous utilisez une tension d'alimentation de plus de 12V DC. Avec une alimentation 12 V DC, ceci permet l'alimentation du régulateur de 5 V de la carte.	
4	Connectez votre tension d'alimentation du moteur ici, maximum de 35V DC. Si U > 12V il faut impérativement retirer le cavalier 3 et alimenté la borne 6 avec une tension de 5V DC venant par exemple d'un Arduino	
5	GND	
6	Sortie 5V si le cavalier 3 (12V) est en place et alimente la logique 5V du circuit	
7 et 12	Pour un moteur à courant continu c'est un cavalier de validation. Il permet le fonctionnement de ce moteur si le cavalier est mis. Sans le cavalier et si c'est relié à une sortie PWM DC cela permettra de contrôler la vitesse du moteur. Pour un moteur pas à pas laissez-le en place.	
8	IN 1	IN1 raccordés à 5V et IN2 à GND le Moteur 1 tournera dans le sens horaire. IN1 raccordés à GND et IN2 à 5V moteur 1 tournera dans le sens Anti-horaire.
9	IN 2	
10	IN 3	Même raisonnement pour le moteur 2
11	IN 4	
13	Sortie 2 Moteur DC "+" ou moteur pas à pas B+	
14	Sortie 2 Moteur DC "-" ou moteur pas à pas B-	

Contrôle Moteur DC :

Pour commander un ou deux moteurs à courant continu c'est assez facile.

Connecter chaque moteur pour les connexions A et B sur le module de L298N. Si vous utilisez deux moteurs qui doivent avoir le même sens de rotation, veiller à ce que la polarité des moteurs soit la même sur les deux moteurs.

Ensuite, connectez votre alimentation :

- le positif à la broche 4 sur le module et le 0V (le négatif pour certain) / GND à la broche 5.
- Si vous fournissez une tension jusqu'à 12V vous pouvez laisser le cavalier « 12V » (point 3 dans l'image ci-dessus) et le 5V sera disponible de la broche 6 sur le module. [Cela peut alimenter à votre Arduino en 5V pour l'alimenter à partir de l'alimentation électrique des moteurs. Ne pas oublier de connecter GND Arduino à la broche 5 sur le module ainsi pour assurer l'alimentation de votre Arduino.](#)

Maintenant, vous aurez besoin de six broches de sortie numériques sur votre Arduino, dont deux doivent être PWM (modulation de largeur d'impulsion). Les broches PWM sont désignés par le symbole ("~") à côté du numéro d'identification, par exemple:



Dans notre exemple, nous avons deux moteurs à courant continu, les broches numériques D9, D8, D7 et D6 seront connectés respectivement aux broches IN1, IN2, IN3 et IN4.

Retirer les deux straps en broche 7 et 12, puis connectez D10 du module à la broche 7 et D5 du module à la broche 12

La direction du moteur est commandé en envoyant un signal élevé ou faible pour chaque moteur.

Par exemple, si pour IN1 est mis au niveau haut et IN2 est mis au niveau bas, le moteur tournera dans le sens horaire, et si IN1 est mis au niveau bas et IN2 au niveau haut va le faire tourner dans l'autre sens.

Cependant, les moteurs ne tournent jusqu'à un MAXIMUM qui est réglé sur la broche d'activation (7 moteur 1, 12 pour moteur 2).

Ils peuvent être arrêtés avec une valeur 0 à la même broche (7 ou 12). Toutefois, si vous avez besoin de contrôler la vitesse des moteurs, le signal PWM de l'Arduino connecté à la broche de validation (7 ou 12) permet cela.

Voici 2 exemples de programmation.

PROGRAMME N°1 :

Faire tourner les deux moteurs à une vitesse imposée et après 2s les faire changer de sens et 2s après les arrêter.

```
// Connecter les entrées du module L 298 N aux bornes suivantes de l'Arduino
// motor 1
int M1 = 10;
int in1 = 9;
int in2 = 8;
// motor 2
int M2 = 5;
int in3 = 7;
int in4 = 6;
void setup()
{
  // Détermination des sorties de l'Arduino
  pinMode(M1, OUTPUT);
  pinMode(M2, OUTPUT);
  pinMode(in1, OUTPUT);
  pinMode(in2, OUTPUT);
  pinMode(in3, OUTPUT);
  pinMode(in4, OUTPUT);
}
void loop()
{
  // Faire tourner les moteurs à une vitesse imposée
  // Faire tourner le moteur 1
  digitalWrite(in1, 1); // 1 indique niveau haut ou HIGH
  digitalWrite(in2, 0); // 0 indique niveau bas ou LOW
  // Vitesse 200 sur un réglage possible de 0 à 255
  analogWrite(M1, 200);
  // turn on motor B
  digitalWrite(in3, 1); // 1 indique niveau haut ou HIGH
  digitalWrite(in4, 0); // 0 indique niveau bas ou LOW
  // Vitesse 200 sur un réglage possible de 0 à 255
  analogWrite(M2, 200);
  delay(2000); // attendre 2000 ms soit 2s
  // Changer le sens de rotation des moteurs
  digitalWrite(in1, 0);
  digitalWrite(in2, 1);
  digitalWrite(in3, 0);
  digitalWrite(in4, 1);
  delay(2000);
  // Arrêter les moteurs
  digitalWrite(in1, 0);
  digitalWrite(in2, 0);
  digitalWrite(in3, 0);
  digitalWrite(in4, 0);
}
```

PROGRAMME 2 :

Faire tourner les deux moteurs à une vitesse qui s'accélère de 0 à max puis diminuer la vitesse des deux moteurs et couper leur alimentation.

```
// Connecter les entrées du module L 298 N aux bornes suivantes de l'Arduino
// motor 1
int M1 = 10;
int in1 = 9;
int in2 = 8;
// motor 2
int M2 = 5;
int in3 = 7;
int in4 = 6;
void setup()
{
  // Détermination des sorties de l'Arduino
  pinMode(M1, OUTPUT);
  pinMode(M2, OUTPUT);
  pinMode(in1, OUTPUT);
  pinMode(in2, OUTPUT);
  pinMode(in3, OUTPUT);
  pinMode(in4, OUTPUT);
}
void loop()
{
  // Autoriser l'alimentation des deux moteurs dans le même sens
  digitalWrite(in1, 0);
  digitalWrite(in2, 1);
  digitalWrite(in3, 0);
  digitalWrite(in4, 1);
  // Accélérer la vitesse des moteurs de 0 à maxi
  for (int i = 0; i < 256; i++)
  {
    analogWrite(M1, i);
    analogWrite(M2, i);
    delay(20); // délai de 0.02 s entre change changement de vitesse
  }
  // Diminuer la vitesse des moteurs de max à 0
  for (int i = 255; i >= 0; --i)
  {
    analogWrite(M1, i);
    analogWrite(M2, i);
    delay(20);
  }
  // Arrêter l'alimentation des moteurs
  digitalWrite(in1, 0);
  digitalWrite(in2, 0);
  digitalWrite(in3, 0);
  digitalWrite(in4, 0);
}
```